

NETWORK CONFIGURATION OF ORACLE AND DATABASE PROGRAMMING USING SQL

Melton Davis¹, Jibril Abdurrashid^{2s}, Philip Diaz³ and Prof. W. C. Harris³

¹Department of Computer Science, City College of New York (CUNY)

²Department of Computer Information Systems, Baruch College (CUNY)

³Department of Physical, Environmental and Computer Sciences,
Medgar Evers College of the City University of New York

I. ABSTRACT:

A database can be defined as a collection of information organized in such a way that it can be retrieved and used. A database management system (DBMS) can further be defined as the tool that enables us to manage and interact with the database. The Oracle 8 Server is a state-of-the-art information management environment. It is a repository for very large amounts of data, and gives users rapid access to that data. The Oracle 8 Server allows for sharing of data between applications; the information is stored in one place and used by many systems. My research will focus primarily on SQL (Structured Query Language) programming. SQL is the way you define and manipulate data in Oracle's relational database. SQL is the industry standard adopted by all database vendors. When programming with SQL, you work on sets of data (i.e., information is not processed one record at a time).

II. BACKGROUND

A little background on the evolution of databases and database theory will help you understand the workings of SQL. Database systems store information in every conceivable business environment. From large tracking databases such as airline reservation systems to a child's baseball card collection, database systems store and distribute the data that we depend on. Until the last few years, large database systems could be run only on large mainframe computers. These machines have traditionally been expensive to design, purchase, and maintain. However, today's generation of powerful, inexpensive workstation computers enables programmers to design software that maintains and distributes data quickly and inexpensively.

The most popular data storage model is the relational database, which grew from the paper "A Relational Model of Data for Large Shared Data Banks," written by Dr. E.F.Codd in 1970. SQL evolved to service the concepts of the relational database model. Most databases have had a "parent/child" relationship; that is, a parent node would contain pointers to its children. This method has several advantages and many disadvantages. In its favor is the fact that the physical structure of data on a disk becomes unimportant. The programmer simply stores pointers to the next location, so data can be accessed in this manner. Also, data can be added and deleted easily. However, different groups of information could not be easily joined to form new information. The format of the data on the disk could not be arbitrarily changed after the database was created. Doing so would require the creation of a new database structure.

Codd's idea for an RDBMS uses mathematical concepts of relational algebra to break down data into sets and related common subsets.

Because information can naturally be grouped into distinct sets, Dr. Codd organized his database system around this concept. Under the relational model, data is separated into sets that resemble a table structure. This table structure consists of individual data elements called columns or fields. A single set of a group of fields is known as a record or row.

When programming, you can embed SQL statements into your source code using an editor of your choice. This source code is then used as input for the precompiler, which in turn translates all the embedded SQL statements into native code. Finally, you can compile and link this code to create an executable after including some Oracle calls.

III. SQL, SQL*PLUS, PL/SQL:

SQL, SQL*Plus, PL/SQL: What's the difference? This is the most often asked question for people new to Oracle. There are several products with the letters "SQL" in the title, and these three, SQL*Plus, SQL, and PL/SQL, are often used together. Because of this, it's easy to become confused as to which product is doing the work and where the work is being done. I will now briefly describe each of these three products.

SQL stands for Structured Query Language. This has become the lingua franca of database access languages. It has been adopted by the International Standards Organization (ISO) and has also been adopted by the American National Standards Institute (ANSI). When you code statements such as SELECT, INSERT, UPDATE, and DELETE, SQL is the language you are using. It is a declarative language and is always executed on the database server. Often you will find yourself coding SQL statements in a development tool, such as PowerBuilder or Visual Basic, but at runtime those statements are sent to the server for execution.

PL/SQL is Oracle's Procedural Language extension to SQL. It too, usually runs on the database server, but some Oracle products such as Developer/2000 also contains a PL/SQL engine that resides on the client. Thus, you can run your PL/SQL code on either the client or the server depending on which is more appropriate for the task at hand. Unlike SQL, PL/SQL is procedural, not declarative. This means that your code specifies exactly how things get done. As in SQL, however, you need some way to send your PL/SQL code up to the server for execution. PL/SQL also enables you to embed SQL statements within its procedural code.

SQL*Plus is an interactive program that allows you to type in and execute SQL statements. It also enables you to type in and execute PL/SQL code and is one of the most common front ends used to develop and create stored PL/SQL procedures and functions.

This tight-knit relationship between PL/SQL, SQL, and SQL*Plus is the cause for some of the confusions between the products. What happens when you run SQL*Plus and type in a SQL statement? Where does the processing take place? What exactly does SQL*Plus do, and what does the database do? If you are in a Windows environment and you have a database server somewhere on the network, the following things happen:

- 1) SQL*Plus transmits your SQL query over the network to the database server.
- 2) SQL*Plus waits for a reply from the database server.
- 3) The database server executes the query and transmits the results back to SQL*Plus.
- 4) SQL*Plus displays the query results on your computer screen.

Even if you're not running in a networked Windows environment, the same things happen. The only difference might be that the database server and SQL*Plus are running on the same physical machine. This would be true, for example, if you were running Personal Oracle on a single PC.

PL/SQL is executed in much the same manner. Type a PL/SQL block into SQL*Plus, and it is transmitted to the database server for execution. If there are any SQL statements in the PL/SQL code, they are sent to the server's SQL engine for execution, and the results are returned back to the PL/SQL program.

The important thing is that SQL*Plus does not execute your SQL queries. SQL*Plus also does not execute your PL/SQL code. SQL*Plus simply serves as your window into the Oracle database, which is where the real action takes place.

One last comment, a special type of index supported by many database systems allows the database manager or developer to cluster data. When a clustered index is used, the physical arrangement of the data within a table is modified. Using a clustered index usually results in faster data retrieval than using a traditional, nonclustered index.

IV. Future Work:

I have created tables using SQL on the Physical, Environmental and Computer Science Department's Database Management System. The goal of this research will be to implement database applications that will be utilized by the department. The applications will range from information about the department to storing scientific data.

Acknowledgements:

This project is supported by NASA MUSPIN CCNY/CUNY NRTS, NASA-GISS/MEC Partnership and NSF NYC-LSAMP

References:

1. Michael Abbey. *Oracle8 The Ultimate Introduction Oracle Press Edition*, 1997, McGraw Hill
2. Lynnwood Brown. *Oracle Database Administration on UNIX Systems*, 1997, Prentice Hall